

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 7,062,759 B2
APPLICATION NO. : 09/839910
DATED : June 13, 2006
INVENTOR(S) : Robison

Page 1 of 5

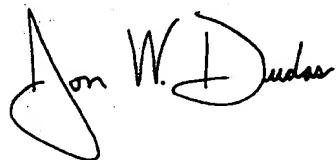
It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Delete Title page illustrating a figure(s), and substitute therefor, new Title page illustrating a figure(s). (attached)

Delete drawing sheet 1,3 and 4A, and substitute therefor drawing sheet 1-,3 and 4A. (attached)

Signed and Sealed this

Fourth Day of September, 2007



JON W. DUDAS
Director of the United States Patent and Trademark Office

(12) United States Patent
Robison(10) Patent No.: US 7,062,759 B2
(45) Date of Patent: *Jun. 13, 2006(54) METHOD AND SYSTEM FOR
INTERPROCEDURAL SIDE EFFECT
ANALYSIS

(75) Inventor: Arch Robison, Champaign, IL (US)

(73) Assignee: Intel Corporation, Santa Clara, CA
(US)(*) Notice: Subject to any disclaimer, the term of this
patent is extended or adjusted under 35
U.S.C. 154(b) by 717 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: 09/839,910

(22) Filed: Apr. 19, 2001

(65) Prior Publication Data

US 2004/0015903 A1 Jan. 22, 2004

(51) Int. Cl.
G06F 9/43 (2006.01)(52) U.S. Cl. 717/141; 717/155; 717/151;
717/154; 717/157; 717/141; 717/156(58) Field of Classification Search 717/159,
717/4, 160, 133, 140-161

See application file for complete search history.

(56) References Cited

U.S. PATENT DOCUMENTS

5,485,616 A *	1/1996	Burks et al.	717/133
5,671,419 A *	9/1997	Carini et al.	717/145
5,790,866 A *	8/1998	Robison	717/160
6,173,444 B1 *	1/2001	Archambault	717/159
6,427,234 B1 *	7/2002	Chambers et al.	717/140
6,721,945 B1 *	4/2004	Sinha	717/157
6,820,253 B1 *	11/2004	Robison	717/141
2002/0010911 A1 *	1/2002	Cheng et al.	717/4

OTHER PUBLICATIONS

Hind et al., Interprocedural Pointer Alias Analysis, Jtd.
1999, ACM.*Cooper et al., Interprocedural Side-Effect Analysis in Linear
Time, Jun. 1988, ACM.*Cooper et al., The Impact of Interprocedural Analysis and
Optimization in the Rn Programming Environment, 1986,
ACM.*Cheng et al., A Practical Interprocedural Pointer Analysis
Framework, Apr. 1999, Univ. of Illinois.*Atkinson et al., Effective Whole-Program Analysis in the
Presence of Pointers, 1998, ACM.*Lakhota, Constructing call multigraphs using dependence
graphs, 1993, ACM.*Liu et al., Eliminating Two Kinds of Data Flow Inaccuracy
in the Presence of Pointer Aliasing, 1997, IEEE.*Harrold et al., Separate Computation of Alias Information
for Reuse, 1996, IEEE.*Boyer et al., Efficient Implementation of Lattice Operations,
ACM, 1989.*

(Continued)

Primary Examiner—Kakali Chaki

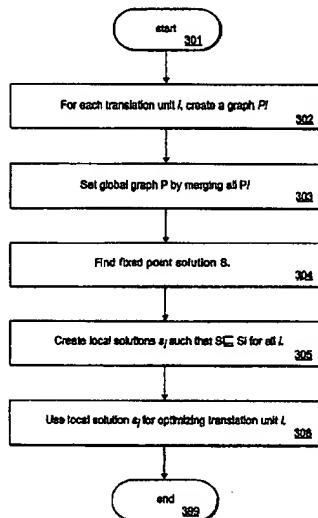
Assistant Examiner—Jinsun Kang

(74) Attorney, Agent, or Firm—Blakely, Sokoloff, Taylor &
Zafman LLP

(57) ABSTRACT

Interprocedural side-effect analysis is performed by constructing a fixed-point problem graph for each translation unit of a software program having a plurality of separately compilable components. The method performs analyzing each routine, of a software program having a plurality of separately compilable routines, to create a plurality of local side-effect problems for each routine; and merging the local side-effect problems to create a global side-effect problem.

18 Claims, 13 Drawing Sheets



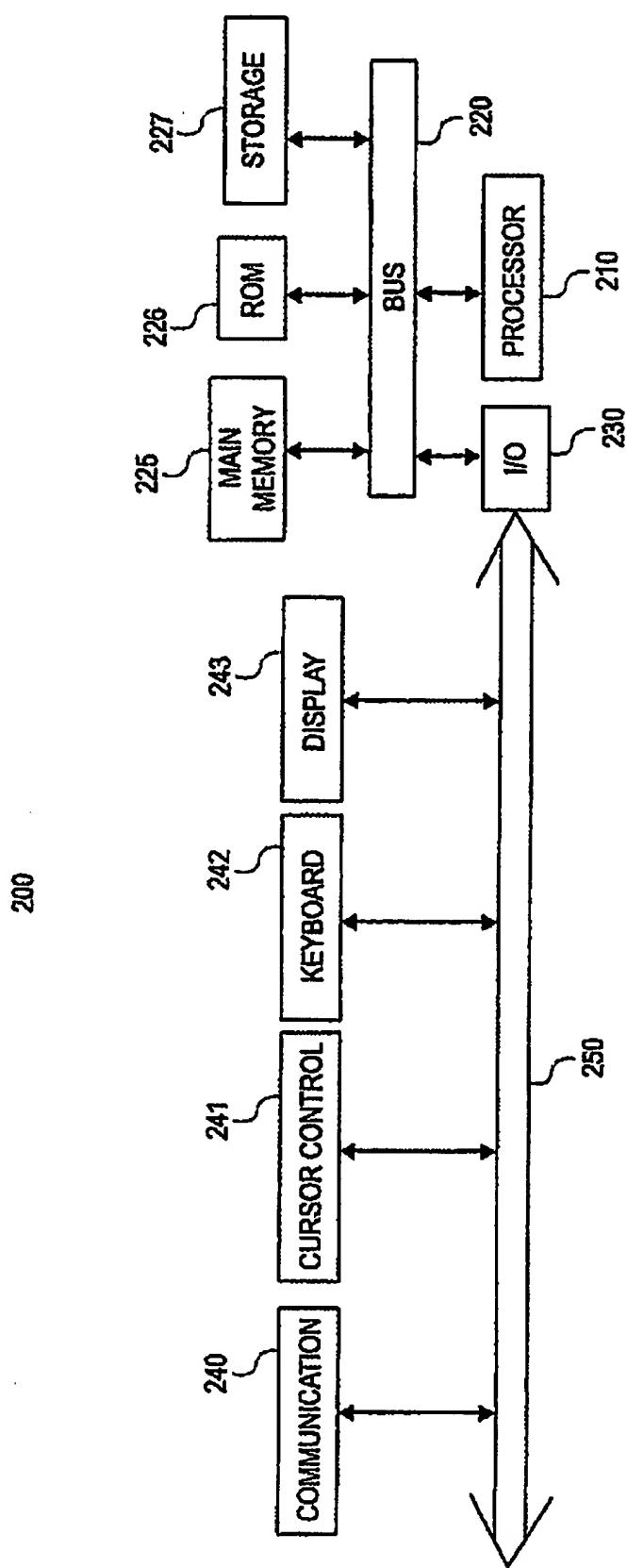


FIG. 1
(PRIOR ART)

U.S. Patent

Jun. 13, 2006

Sheet 3 of 13

7,062,759 B2

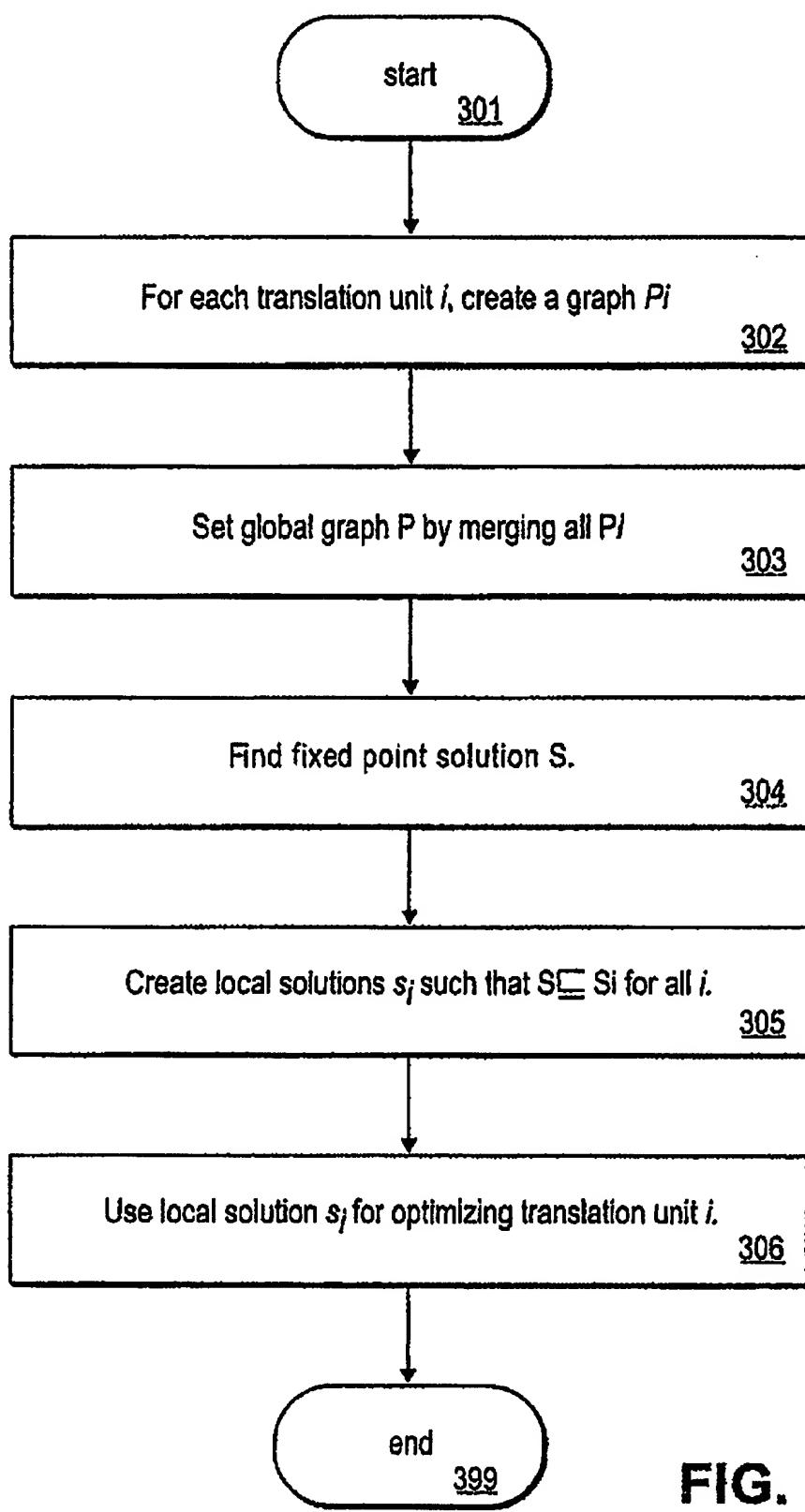


FIG. 3

U.S. Patent

Jun. 13, 2006

Sheet 4 of 13

7,062,759 B2

Function	Function (x,y)
410 ~ TOP	(PURE,PURE)
420 ~ COPY	(y,y)
430 ~ IN_TO_LOST	if $y \leq l \Rightarrow (\text{LOST},\text{LOST})$ otherwise $\Rightarrow (\text{PURE},\text{PURE})$
440 ~ UNRETURN	
450 COPY_AND_IN_TO_LOST	if $y = \text{LOST} \Rightarrow (\text{LOST},\text{LOST})$ otherwise $\Rightarrow (z,z)$ where $z = y \sqcup OI$
460 ~ CAT_FORMAL	if $y \leq l \Rightarrow (\text{LOST},\text{LOST})$ otherwise $\Rightarrow (y,y)$ (y,PURE)
470 ~ CAT_ACTUAL	(PURE,y)
480 ~ GATE	if $x = \text{LOST} \Rightarrow (\text{LOST},\text{LOST})$ else if $x < R$ (z,z) where $z = (x \sqcup OI) \sqcap y$ else (z,z) where $z = (x \sqcup OI)$

FIG. 4A